

Reconfigurable data encoding schemes for on-chip interconnect power reduction in deep submicron technology

Vennapusapalli Shavali¹, Sreeramareddy Gorlagummanahally Maripareddy², Patil Ramana Reddy¹

¹Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Anantapur, Anantapur, India

²Department of Electronics and Communication Engineering, C. Byregowda Institute of Technology, Kolar, India

Article Info

Article history:

Received May 17, 2022

Revised Aug 5, 2022

Accepted Sep 13, 2022

Keywords:

Coupling transitions

Interconnects

Low power

Network on chip

Power dissipation

Self-transitions

ABSTRACT

With technology scaling, size of both transistor and interconnects are reduced. Power dissipation due to dynamic switching is high in the interconnects. Suitable encoding schemes that reduce transition between data bits are used to minimize interconnect power dissipation. In this paper transition between data bits is minimized based on three novel data encoding schemes identifying the novel methods estimates bit transitions in a pair of data bits and performs half inversion or full inversion on one byte of data thus reducing the switching activity by 50%. The encoder and decoder for the three encoding schemes are modelled in verilog hardware description language (HDL) and implemented using application specific integrated circuit (ASIC) flow targeting 32 nm. Technology over all power dissipation of encoding scheme is 1.04 μ W in addition over head area of 210 cells with encoding delay of 340 ps. Encoder decoder register transfer logic (RTL) code is implemented and the total area required is 34980 units. The data encoding and decoding schemes are suitable for low power applications.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vennapusapalli Shavali

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Anantapur (JNTUA)

Anantapur, A. P, India

Email: v.shavali@gmail.com

1. INTRODUCTION

Cores require network on chip (NOC). Multicore processors or many core processors provide improvement in performances for computing activity with trade-off between power and frequency. The performance and power consumption of multicore technology are limited by interconnect fabric and cache coherence protocols. Three alternatives for high-speed interconnects are wireless NOC, radio frequency (RF) interconnects and surface wave interconnects. RF interconnects [1] and optical interconnects [2] are also been reported to provide alternate solutions for on-chip data transmission. The Zenneck surface wave (SW) is another method used for wireless transmission of data on-chip [3]. Next generation high performance computing based on wired interconnects have been discussed in [4] reducing power, longer wire issues and improving low bandwidth considering 3D mesh interconnect architecture. Use of wired interconnects for multi-core on-chip communication is still provides better solution due to its simplicity and cost. Buses are used to connect multi-core chips but scalability of interconnects have limited their use for multi-core interconnection. Longer wires consume more energy as compared with short wires considering same current flow. With technology scaling size of interconnects are reduced and higher density in interconnection is achieved with 11 to 13 metal layers. Interconnects in each layer have a defined pitch size, minimum size of wire and minimum space between wires with electrical characteristics that the designers have to consider.

The metal layers at the lower layers are used for local interconnects and have narrower width. Metal layer in the higher layer are used for power and clock and have wider widths.

The metal interconnects in the middle layer are used for connecting between multiple cores considering intel-xeon-phi-processor-7290-16gb-1_50-ghz-72-core the internal details considering two cores being connected in metal 4 and metal 5 interconnects it is estimated that there will be 101,520 wires equal to transmitting 128 bits using 793 unidirectional bus interconnects. The energy cost of single data transfer or also called as flit is the count of bit toggles that occurs on the wire. The activity factor or flit depends upon on the data being transmitted and the previous data transmitted [5] compression of data is used to remove redundancy in data links [6] to improve system performance and energy consumption. Compression of data however has overheads such as latency in data transmission, additional area and power dissipation due to compression/decompression circuit logic, complexity and cost [7]. In the studies carried by Pekhimenko *et al.* [8] at compression schemes have been presented for energy-efficiency of communication based on novel toggle aware compression technique. Encoding of data prior to transmission over interconnects have demonstrated decrease in energy consumption in the interconnects and have been successfully utilized for NOC technology [9]. The work reported in [10] have used reversible logic to encode the data prior to transmission over interconnect to reduce power dissipation and area overheads. Survey in [11] have conducted study on the factors that affect interconnect delay such as distance and bandwidth of interconnects, toggle rate, voltage and frequency of data travelling over interconnect considering advance micro device (AMD) graphic processing unit (GPU) designed using 28nm technology. Transmitting zeros or ones consume very small power and transmitting zeros consumes more power than ones. Power dissipation influenced by data travelling in nearby wire has very less impact but data toggle has significant impact on interconnect power dissipation. The data movement over interconnect at different voltage and frequency (considering DVFS as V^2f) increases power dissipation as either of V or f increases. Interconnect bandwidth also impact power dissipation and it is observed that with data rate change from 210 GB/s to 104 GB/s the interconnect power is reduced by 50%.

The work reported by Wang *et al.* [11] use differential coplanar waveguide (CPW) for on-chip data transmission with waveguide of 55- μm pitch size and pulse amplitude modulation (PAM) signalling is used for data transmission demonstrating 10x improvement in bandwidth density of 6.1 Gb/s/ $\mu\text{m.cm}$. Low swing signalling and bus coding techniques have been used to reduce power dissipation. The techniques proposed by Zhang *et al.* [12] recommends use of extra power supply, multi threshold technology and multiple interconnects along with differential signalling to reduce interconnect power dissipation. The work reported by Chaudhary *et al.* [13] have used low swing voltages to drive signal on long wire interconnects demonstrating power reduction. Garcia *et al.* [14] high speed adaptive mj-driver for driving global interconnect lines is proposed with specific range of capacitance at 500 Meha hertz frequency. Use of multiple paths for driving signal from source to destination is presented. Postman and Chiang [15] reduced recurrent minimum power encoding (RRMP) encoding schemes presented. Chennakesavulu *et al.* [16] estimates data correlation between each bit in the data stream to be transmitted over interconnect. The RRMP scheme addresses reduction in power dissipation by 21 mW operating at 753.43 MHz on Virtex-7 FPGA. The memory bits stored in synchronous random access memory will consume power for dedicated routing utilization by selection of any of the routing resources. Two new encoding schemes to address self-coupling and switching activity is presented in [17] even-odd-full (EOF) inversion schemes and EOF scheme with segmentation have been developed and implemented using 45nm technology and, in their work, and energy, delay and energy-delay efficiency is computed for 8-bit to 64-bit is computed. Bus invert (BI) method is presented in [18] for encoding the data stream prior to transmission over interconnect reducing power dissipation and is evaluated for memory module. Nearest neighbor algorithm is used for computing optimal reordering algorithm and an adaptive word reordering method is developed for reducing power dissipation in [19].

The proposed scheme is evaluated in data transfer in direct memory access module and more than 30% of power saving is demonstrated. An adaptive word reordering (AWR) scheme has been developed in [20] that adaptively reduces the signal transitions leading to power dissipation. The power dissipation due to AWR scheme is compared with other schemes such as BI, APBI and ABE with interconnect power dissipation reduced by 50% compared to all other schemes. Code inversion encoding and decoding schemes are presented to improve the cross-point phase change memory (PCM) read margin in [20] sneak current is increased to reduce read margin lowering static cell resistance that improves read margin of the PCM. Different model parameters can be derived for 32 nm, 22 nm, and 10 nm technology by scaling and power is estimated in [21]. Temperature variations for interconnection delay and leakage power dissipation for encoding system is presented in [22]. Encoding techniques such as bus invert logic proposed by Stan and Burleson [23] inverts the data if more than half of the bus lines are activated for a given data to be transmitted. Along with the data additional bus that carries the inv-bit information is included. Many versions of bus invert logic are presented in [2]-[7] and all these methods use additional bus lines for data

transmission. Pekhimenko *et al.* [8] selectively activated flip-driver (SAFD) method uses a flip driver for sending data demonstrating 35% of by transitions. Yoon [24] has proposed a two-bit bus invert coding scheme for low power design of very large scale integrated (VLSI) circuit to reduce interconnect power dissipation and in this work, the bus transitions have been reduced. Cross-coupling between interconnects contributes to power dissipation and to compute the interconnect power it is required to consider several factors. Xie *et al.* [25] presented a technique for power reducing in network on chip switching activity is reduced by examining the handover transition and coupling between the links to reduce the power consumption and cross coupling. using. Sangewar [26] concentrated chip link and router link and minimizes its own coupling effect of capacitance.

Most of the studies focus on developing coding methods and reducing of area and power dissipation of add-on logic for data transmission. It is required to consider design trade-off between both encoding, area overhead and power for efficient on-chip data communication. In this paper, new coding algorithm for data transmission is presented. Area, power and speed performances of proposed logic are studied considering in three different technologies considering full chip design flow. Section 2 presents a detailed discussion on proposed method for data encoding to minimize power dissipation, section 3 presents discussion on application specific integrates circuit (ASIC) implementation and results are presented in section 4 with conclusion in section 5.

2. PROPOSED METHOD

Power dissipation in an integrated circuit is not only due to the logic circuits but also interconnects. As the data travels over interconnects the data switching or transition from ‘1’ to ‘0’ or from ‘0’ to ‘1’ leads to power dissipation over interconnects. A detailed discussion is presented in the previous section on the relation between power dissipation and interconnects. Coupling transitions that occur between adjacent bus lines during logic level transitions need to be addressed with reducing the number or having common transitions over adjacent lines. Self-transitions are also to be considered and minimized by having minimum number of transitions from previous logic to present logic. Another major power dissipation factor is the width of bus lines. In order to reduce power dissipation, it is required to reduce the number of bus lines travelling long distances over the chip. In order to minimize power dissipation in long lines and bus lines new method is presented in this work.

The encoding schemes for reducing the number of transitions of logic over interconnect bus lines are presented in Figure 1. The data bits that need to be transmitted over interconnect from one sub system to next subsystem within an integrated circuit is represented as $a_0, a_1, a_2, \dots a_{N-1}$. The 1:2 de-multiplexer at the input side is controlled to load the input sequences of successive 8-bit data into two 8-bit registers. The two groups of 8-bit data to be transmitted successively over the interconnect are denoted as $b_0, b_1, b_2, \dots b_{M-1}$ and $c_0, c_1, c_2, \dots c_{M-1}$. The first 8-bit data is transmitted over the bus line without any changes in the logic levels. The second or successive 8-bit data is encoded considering the logic levels that were transmitted over previous time interval. The encoding scheme proposed in this work, encodes the data bits to reduce self-transition between two groups of 8-bit data. Along with the 8-bit data ($d_0, d_1, d_2, \dots d_7$) two additional bits (e_0, e_1) are appended and transmitted from the source.

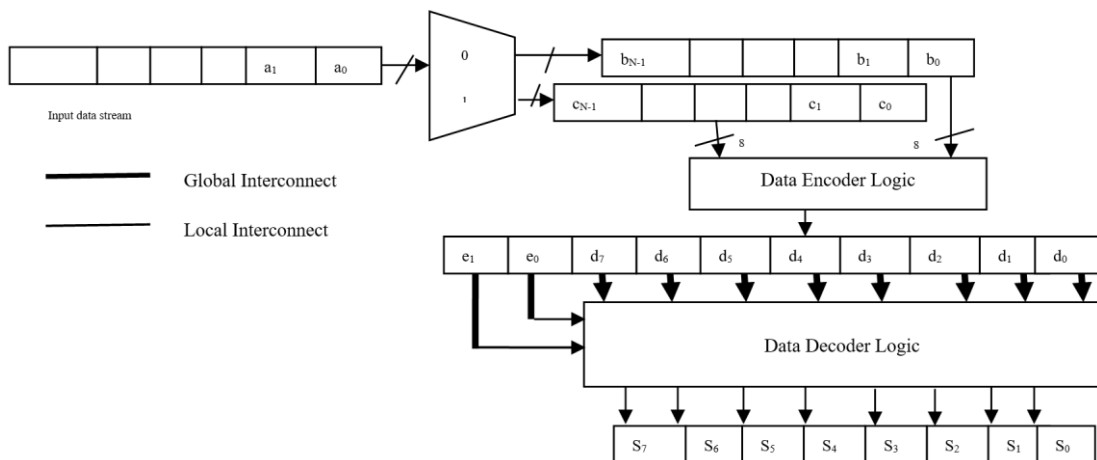


Figure 1. Encoding scheme reducing coupling transition over bus lines

At the destination, the two additional bits are used to decode the encoded 8-bit data (s0, s1, s2, ...,s7). Encoding the data prior to transmission to minimize self-transition requires to address two major challenges: first the number of self-transitions between two groups of 8-bit data need to be minimized, second the encoding and decoding logic at the source and destination respectively need to be realized with minimum number of logic circuits, propagation delay and power dissipation. In this work, the encoder and decoder logic are designed considering all these constraints.

2.1. Design of encoding scheme

Figure 2 presents novel encoding scheme proposed in this work. The two group of 8-bit data is divided into four groups of two-bits and each of the group of data are processed in parallel to generate the final 8-bit data that is transmitted to reduce self-transition. The first stage of processing unit is the transition level logic (TLL) that determines the number of transitions between the two pairs of data and generates two outputs T_1^n and T_0^n to distinguish between transitions (four possible transitions). In addition there are two more bits generated X_1 and X_0 to differentiate between transitions. The 2:4 encoders are used to distinguish between four possible transitions. The third stage sub system is the logic unit that counts the number of times the four possible transitions have occurred in the two groups of 8-bit data which is group logic count unit (GOLCU). The fourth stage and the last stage of the proposed encoding model is the data encoding logic that is designed to generate the data bits that will have minimum number of self-transitions between two successive groups. The last stage has two control inputs R1 and R0 to configure the encoder to generate the data bits with three different schemes {Scheme 1, Scheme 2 and Scheme 3} addressing self-transition. A detailed discussion on design of each subsystem is presented.

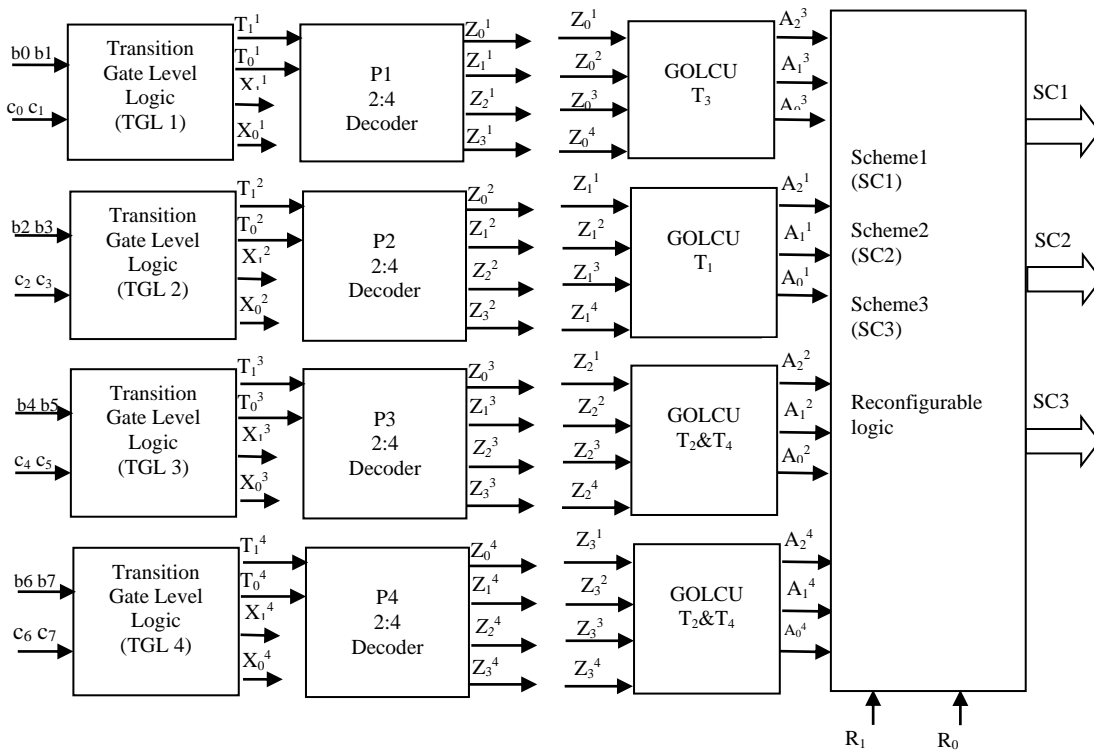


Figure 2. Encoder architecture

2.2. Transition level logic

The transition level logic unit is designed to identify the four possible transitions that can occur between pair of two-bit input Table 1 presents the possible transitions that are captured in TLL. The input bits b0 and b1 are compared with C0 and C1 to generate four outputs T_1^n , T_0^n and X_1 and X_0 . There are four possible transitions with hamming distance 0, 1 and 2 and these transitions are grouped into four transition logic as in Table 1. The transition level between two pair of data based on hamming distance is grouped into four combinations. In order to distinguish between these four combinations two outputs represented as T_1^n , T_0^n are used. From the combinations it is observed that there are two groups with same hamming distance. In

order to distinguish between these two combinations two additional outputs are generated as X_1 and X_0 . If $X_1 X_0$ is 10 it indicates hamming distance of 2, self-transition of 2 with zero coupling transition. If $X_1 X_0$ is 01 it indicates hamming distance of 2, self-transition of 1 with 2 coupling transitions. Based on the TLL it is required to count the number times the TLLs have occurred in the four groups of 8-bit data. Figure 3 represents the circuit diagram for TLL1 logic that is designed to identify transition logic in first pair of 8 bit data. The logic circuit is realized using two stage gate level logic. To compute the four outputs of TLL1 2-4 inputs basic gates and exclusively-OR (XOR) gates are required that introduces non-uniform gate level netlist.

Table 1. Transition level combinations

Present logic level	Next logic level	Hamming distance	Group	$T_1^p T_0^p$	$X_1 X_0$
00	00	0	0	00	00
11	11	0	0	00	00
01	10	2	2	10	01
10	01	2	2	10	01
00	01	1	1	01	00
00	10	1	1	01	00
01	11	1	1	01	00
01	00	1	1	01	00
10	11	1	1	01	00
10	00	1	1	01	00
11	10	1	1	01	00
11	01	1	1	01	00
00	11	2	3	10	10
11	00	2	3	10	10

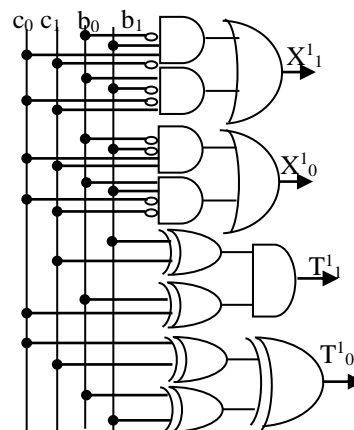


Figure 3. TLL1 logic circuit

2.3. Counting of TLL

In order to count the number of TLLs the proposed logic structure is designed by combining a 2:4 decoder and customized logic unit. The decoder unit generates four outputs Z_0, Z_1, Z_2 and Z_3 indicating whether the input to the decoder belongs to group 0, 1, 2 or 3. Z_0 is 1 if it is group 0, Z_1 is 1 if it is group 1, Z_2 is high if it is group 2 and Z_3 is high if it is group 3. In order to count the number of times group 0 transitions have occurred in the four groups of 8-bits the Z_0 output of all four decoders are set as input to group 0 logic count unit. To count the number of times group 1 transition, occur, Z_1 output of every decoder is set as input to group 1 logic count unit. Similarly, the outputs Z_2 and Z_3 are connected to group 2 and group 3 logic count units. Table 2 presents the functionality of logic count unit.

The TLL count logic generates three bit output $A_2^3 A_1^3 A_0^3$ that is an indication of number of times group 0, 1, 2 or 3 has appeared in the four group combinations of 8-bit data. With four groups of two-bit data the maximum number of times a particular transition can occur is 4 and hence 3-bits are used to represent the number of counts. Similarly, the output of decoders Z_1, Z_2 and Z_3 are combined together to generate count logic. The count logic is designed and the gate level netlist for the count logic is given in Figure 4. Presents the logic circuit for group '0' logic count unit (GOLCU). The output of decoders is grouped into four groups

as follows: Group 0–($z_0^1, z_0^2, z_0^3, z_0^4$), Group 1–($z_1^1, z_1^2, z_1^3, z_1^4$), Group 2–($z_2^1, z_2^2, z_2^3, z_2^4$) and Group 3–($z_3^1, z_3^2, z_3^3, z_3^4$). Each of the group data bits are processed by four different logic count units. The logic circuit for group ‘0’ is designed as in Figure 5 considering the functionality presented in Table 2. In order to achieve uniformity in propagation delay, fan-in and load capacitance distribution the gate level netlist with fan-in of 2 is used. The number of stages are limited to 4. Further optimization in logic control unit (LCU) is achieved by resource sharing design approach. Based on the design approach for group 0 LCU, gate level circuits for all three LCU’s are designed optimizing area and propagation delay. In order to distinguish between group 2 and group 3 that have similar hamming distances the corresponding 2:4 decoder logic circuit is designed. The two outputs X_0 and X_1 that are generated for TLL logic is used as enable input to 2:4 decoders. Only when the logic X_0 or X_1 is high the corresponding 2:4 decoders are enabled and the outputs generated are used as valid inputs to logic count units.

Table 2. TLL count logic

Z_0^1	Z_0^2	Z_0^3	Z_0^4	A_2^3	A_1^3	A_0^3
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

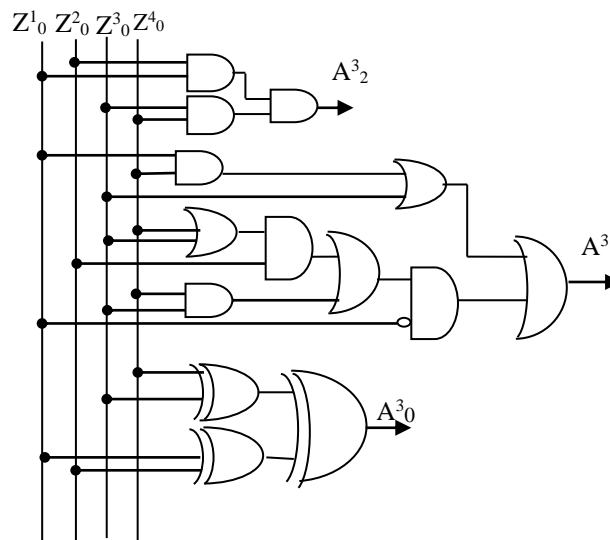


Figure 4. Counting logic diagram

2.4. Design of data inversion scheme

In this work three schemes are proposed for reducing transitions in interconnections. In scheme 1 based on predefined conditions only even bits are inverted and in scheme 2 and 3 all bits are inverted. Figure 5 presents the top-level diagram for generation of control signals for all three schemes. The outputs of logic count units are connected to the schemes conditions control output logic as in Figure 5 each of the control output logic generates outputs EI, FLI, and FL2 representing even inversion for scheme 1, full inversion for scheme 2 and scheme 3. The 3 outputs of schemes control output logic are used to generate full inversion or

even inversion of data bits. Table 3 presents different groups and different levels of transitions which can be classified as different groups.

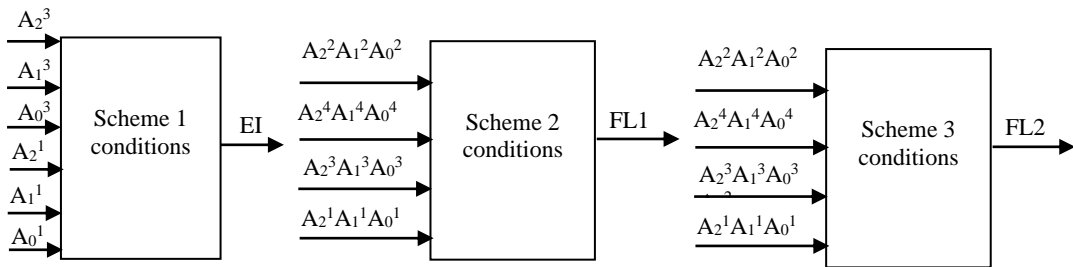


Figure 5. Control logic for three schemes

Table 3. Logic representation

Group	Logic count	Representation
Group 0	$A_2^3 A_1^3 A_0^3$	T_3
Group 1	$A_2^1 A_1^1 A_0^1$	T_1
Group 2	$A_2^2 A_1^2 A_0^2$	T_2
Group 3	$A_2^4 A_1^4 A_0^4$	T_4

The scheme 1 condition unit shown in Figure 5 is designed with a comparator circuit to compare the group 0 and group 1 count outputs. The inputs $\{A_2^3, A_1^3, A_0^3\}$ and $\{A_2^1, A_1^1, A_0^1\}$ are 3-bit inputs representing the count of group 0 and group 1 transitions. The 3-bit comparator is required to find if $T_1 > T_3$ which is designed by considering a two stage 2-bit comparator circuit as shown in Figure 6. The output of 2nd stage is used to generate CI output signal for scheme 1 logic. If CI is high even bits of inputs data is to be inverted else no inversion is carried out. The inversion logic circuit is carried in Figure 7 is used to generate data bits based on scheme 1 logic.

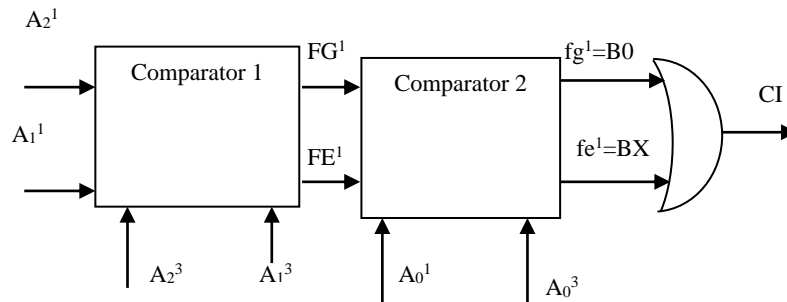


Figure 6. Two stage 2-bit comparator circuit

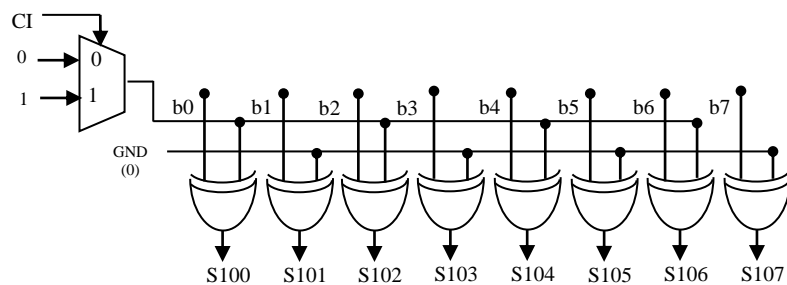


Figure 7. Inversion logic circuit

Figure 8 presents the flow chart for scheme 2 logic proposed in this work. Depending upon either of 3 conditions satisfied full inversion is carried out on the data bits to be transmitted. The circuit diagram for comparing (T_2 and T_4), (T_1 and T_2), (T_1 and T_3) are realized using 2 stage 2-bit comparator circuits. The outputs of all 3 comparators denoted as B0, B1, and B2 are considered for performance full inversion logic shown in Figure 9. The input bits $b_0, b_1, b_2, \dots, b_7$ are inverted based on the control signal CI generated by the scheme2 logic presented in Figure 9.

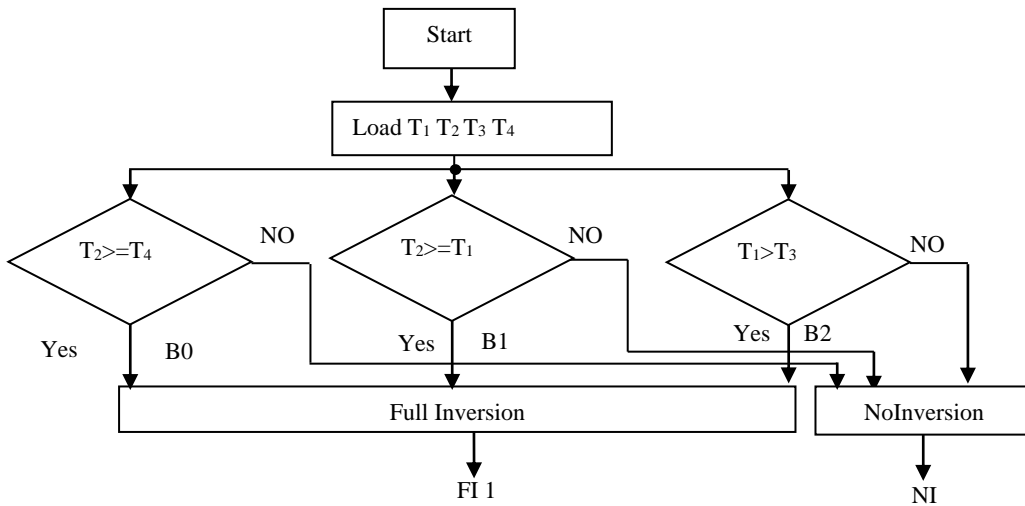


Figure 8. Scheme 2 logic

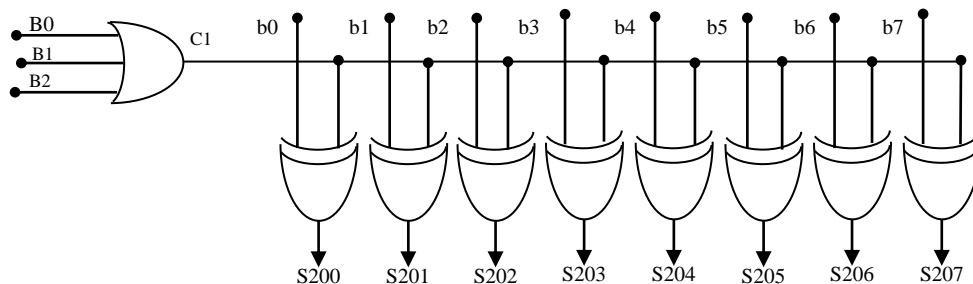


Figure 9. Full inversion logic

Figure 10 presents the flow chart for scheme 3 logic. Considering $T_1, T_2, T_3,$ and T_4 status the input data is fully inverted or not inverted according to the conditions presented and scheme 3 has to satisfies the conditions of scheme 1 that $T_1 > T_3$ and the flow chart shown in Figure 10. The comparison of logic counts are carried out using 2 stage 2-bit comparator circuits. The inversion circuit is similar to the scheme 2 inversion logic. In order to achieve reconfiguration of encoding scheme the outputs generated from scheme 1, 2 and 3 are combined and reconfigurable encoding scheme is proposed. The reconfigurable encoding scheme is shown in Figure 11. Figure 12 presents the design of reconfigurable encoding logic that is designed to perform half inversion, full inversion and no inversion depending upon control inputs I/NV and R. The advantage of the proposed design is that the number of logic circuits required to encode data at the source is optimized considering trade-off between area and delay. The reconfiguration scheme designed is capable of choosing any of the three encoding scheme to minimize self transitions and coupling transitions over the interconnect. The encoded data is transmitted over the interconnect with additional two inputs e_0 and e_1 to distinguish between three encoding schemes. Figure 13 and Figure 14 present the decoder logic to decode the encoded data at the destination for scheme 1 and for scheme 2 use Figure 15 and Figure 16. Scheme 3 is similar to scheme 2.

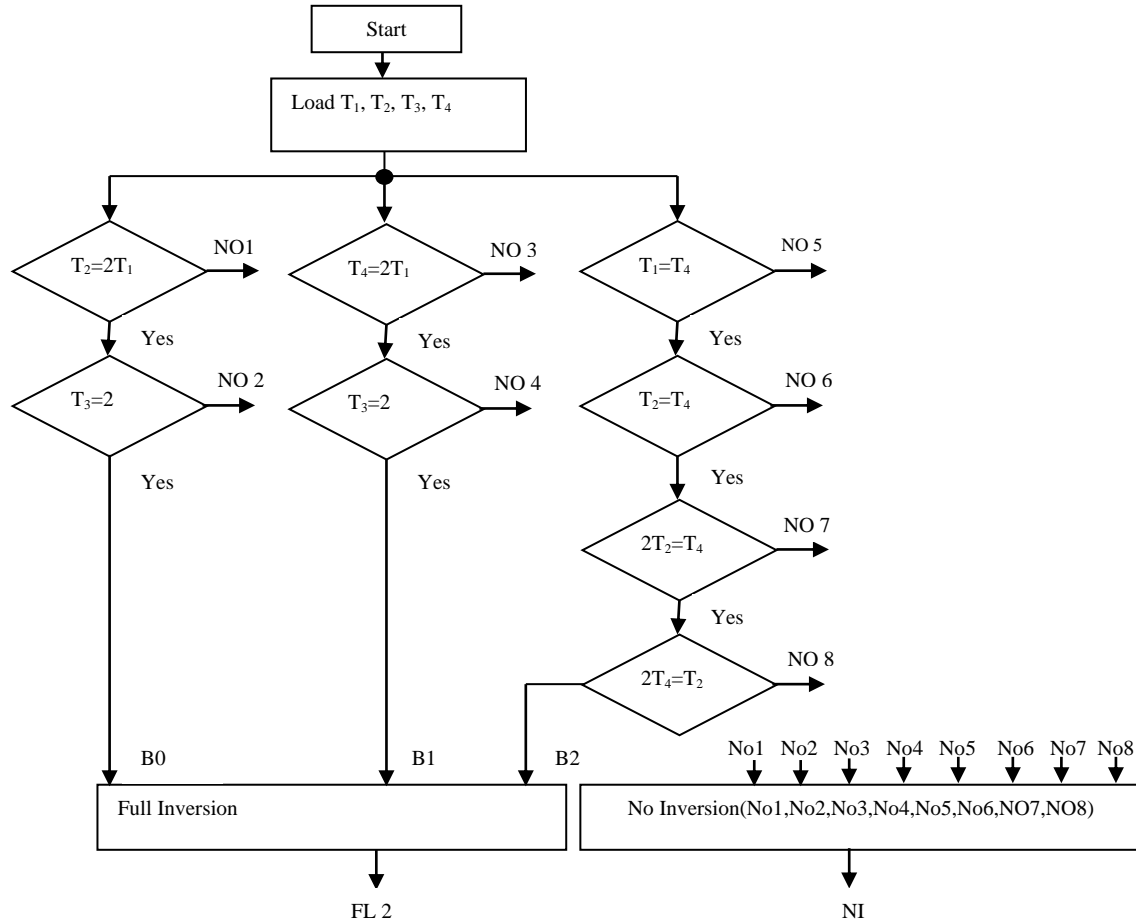


Figure 10. Scheme 3 logic

Figure 12 presents the design of reconfigurable encoding logic that is designed to perform half inversion, full inversion and no inversion depending upon control inputs I / NV and R. The advantage of the proposed design is that the number of logic circuits required to encode data at the source is optimized considering trade-off between area and delay. The reconfiguration scheme designed is capable of choosing any of the three encoding schemes to minimize self transitions and coupling transitions over the interconnect. The encoded data is transmitted over the interconnect with additional two inputs e0 and e1 to distinguish between three encoding schemes. Figures 13 and 14 present the decoder logic to decode the encoded data at the destination for scheme 1 and for scheme 2 use Figure 15 and Figure 16. Scheme 3 is similar to scheme 2.

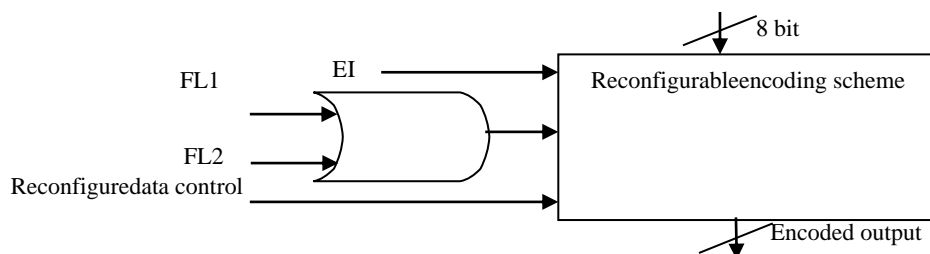


Figure 11. Reconfigurable encoding unit

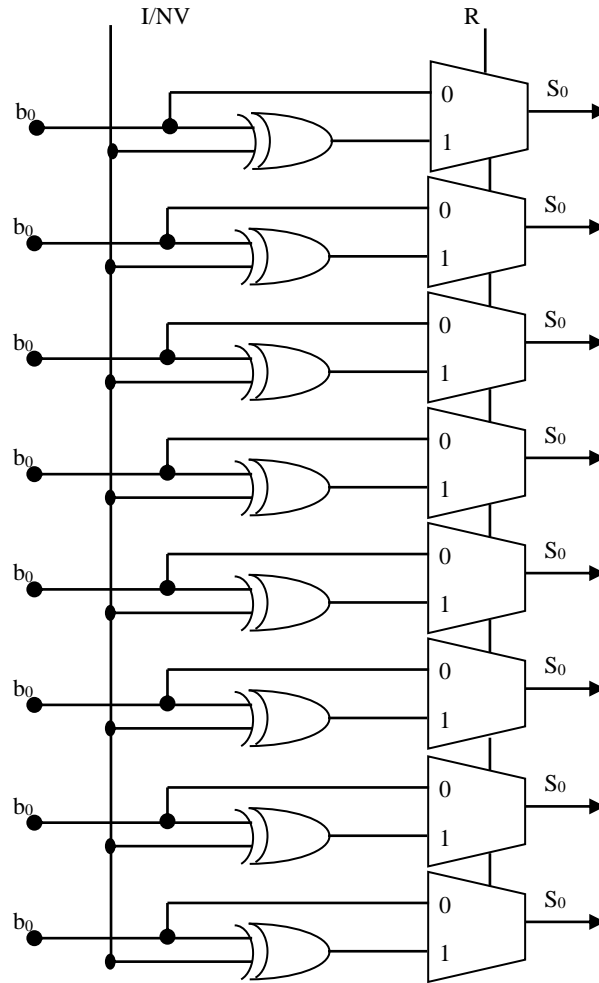


Figure 12. Reconfigurable encoding circuit

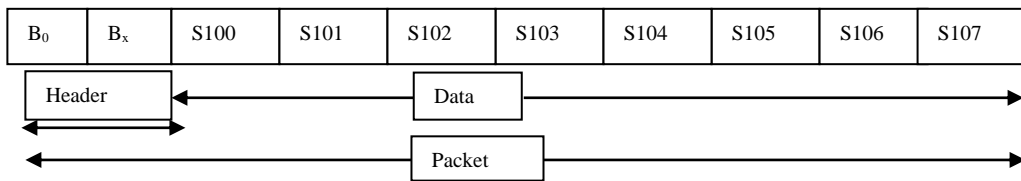


Figure 13. Datapacket of encoding logic for scheme 1

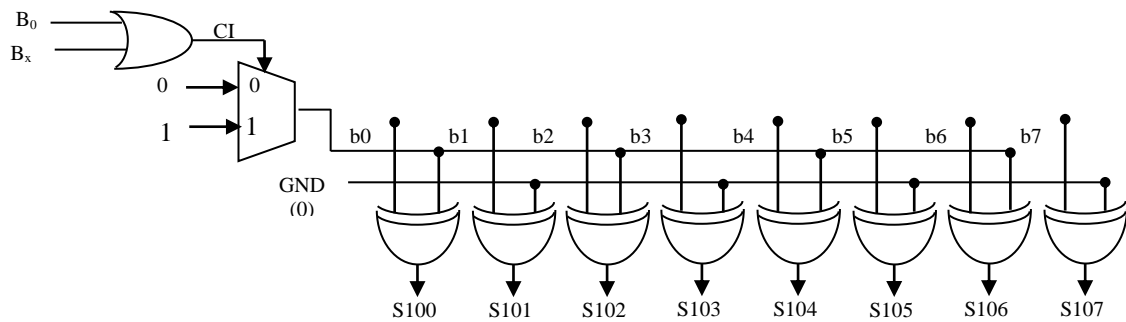


Figure 14. Circuit diagram of decoder for scheme 1

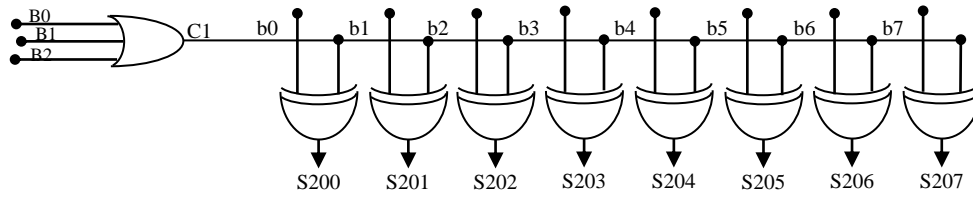


Figure 15. Data packet of encoding logic for scheme 2

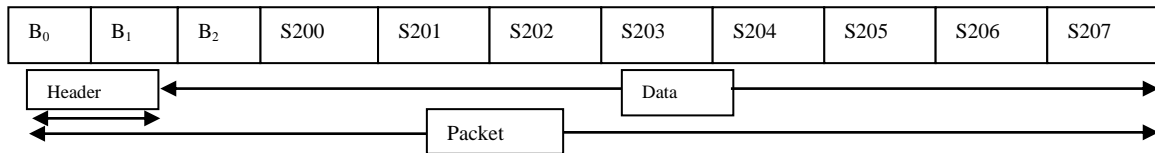


Figure 16. Circuit diagram of decoder for scheme 2

3. ASIC IMPLEMENTATION

The proposed design discussed in section 3 is modelled on verilog HDL and is simulated for its functionality verification. The functionality verification is carried out by considering text vectors of 256 bits. The encoder is verified for its functionality using 256 vectors and decoder outputs are compared to verify logic correctness of the proposed design. The functionally correct encoder and decoder HDL code is synthesized to generate gate level netlist. The ASIC implementation of the gate level netlist is carried out accordingly to the flow diagram shown in Figure 17.

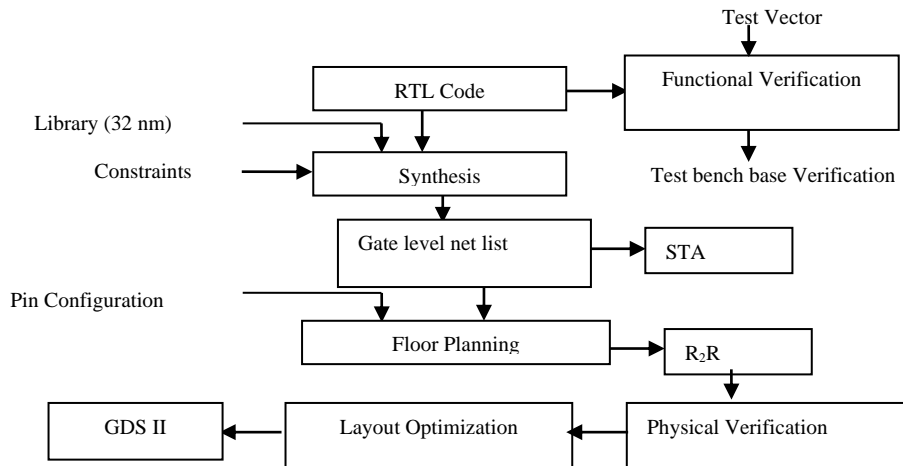


Figure 17. ASIC implement on flow of proposed logic

4. SYNTHESIS AND RESULTS

The HDL code is developed for all the three different schemes considering hierarchical coding methodology. The developed HDL code is verified for its functionality considering test vectors of length 8-bits to 256-bits. Each of the test vectors are grouped to 8-bit and the transition between the two pairs of successive 8-bit is set for both maximum number of transitions with hamming distance of 1, 2, 3 and 4. The groups of 8-bits are also introduced with zero transition. Considering different configuration of transition, the HDL model developed for all three schemes are verified by encoding the data and decoding the data to find out the logic correctness in decoder. The encoder and decoder circuit designed for all three schemes reduces the coupling and self-transition between two consecutive group of data transmitted over interconnect reducing power dissipation. In order to identify the additional complexity introduced by the encoder and decoder, the functionally correct HDL code is synthesized to estimate the gate level count. For synthesis and

ASIC implementation 35 nm complementary metal oxide semiconductor (CMOS) technology is considered and ASIC implementation is carried out using Qflow EDA tool process. Figure 18 presents the ASIC implementation flow considered for evaluation of performance metrics of the proposed design. The RTL code developed is simulated by considering test vectors. The functionally verified design is synthesized to obtain the gate level netlist considering 35 nm library and optimum constraints. Physical design is carried out performing physical verification and final graphic database system information interchange (GDSII) is generated for sign-off. Table 4 presents the summary of gate level count of the encoder-decoder logic. All three schemes are compared with regard to logic gates required for ASIC implementation. Clock constraint is set by defining minimum clock time period of 800 ps, area constraint is set by defining the maximum gate count of 250 and fan-in is defined to maximum of 4.

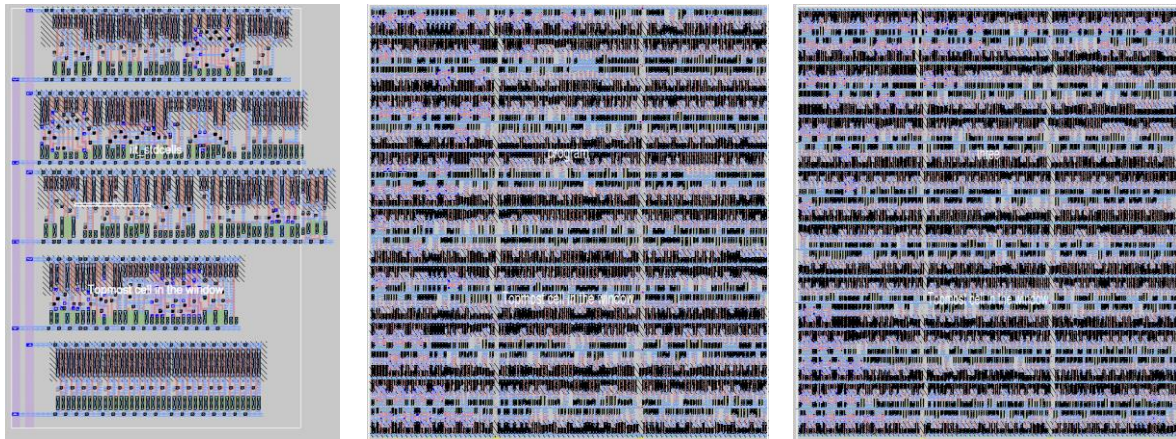


Figure 18. Fully placed, routed GDSII for all 3 schemes

Table 4. Comparison of gate level count for all schemes

Scheme/Gates	SC1	SC2	SC3
Number of wires	125	132	151
Number of wires bits	190	197	218
AND2X2	6	9	4
AOI21X1	10	10	13
AOI22X1	2	2	0
BUF2	18	20	27
DFFPOSX1	32	32	33
INVX1	27	24	19
NAND2X1	18	18	27
NAND3X1	8	8	12
NOR2X1	21	20	22
OAI 21X1	37	34	34
OAI 22X1	8	8	6
XNOR2x1	0	2	2
MUX2X1	0	0	1
OR 2X2	0	0	2
Number of cells	187	187	202

From the synthesis netlist obtained it is identified that for realizing all the three different schemes the gate count is limited to 210. Both scheme 1 and scheme 2 require least number of gate counts and scheme 3 requires 23 additional gates compared with scheme 2 and scheme 1. The 3 input NAND gate with drive strength of X1 and buffers with drive strength of X2 are found to be 12 and 27 for scheme 3 implementation. The number of conditions required to be satisfied in scheme 3 requires additional logic and hence the number of 3 input NAND gates are increased. In order to limit the propagation delay in the sub system or intra connect additionally 7 buffers are included in the gate level netlist. The number of intra connects in scheme 3 is increased by a factor of 17% as compared with other two schemes and the power dissipation in scheme 3 of the encoder-decoder will be increased. static timing analysis (STA) is carried out for all the three schemes and the results are summarized in Table 5. Scheme 3 is found to have minimum propagation delay of 989 ps and operates at maximum frequency of 1011.12 MHz. The minimum path delay in all three schemes is identified between the paths DFFPOSX1_21/CLK to DFFPOSX1_24/D and in scheme 3 with additional

buffers introduced the minimum path delay is reduced by a factor of 17.56% as compared with other two schemes. The maximum delay in scheme 3 is reduced by factor of 53.55% as compared with other schemes. The maximum path delay in scheme 3 is observed between DFFPOSX1_23/CLK to DFFPOSX1_4/D.

Table 5. Static timing analysis

Scheme	Max-Min. Delay	Delay Path	Delay (ps)	Clock Freq. (MHz)
1	Maximum Delay	DFFPOSX1_16/CLK to DFFPOSX1_5/D	2129.46	469.603
1	Minimum Delay	DFFPOSX1_21/CLK to DFFPOSX1_24/D	413.86	469.603
2	Maximum Delay	DFFPOSX1_16/CLK to DFFPOSX1_7/D	2127.98	469.93
2	Minimum Delay	DFFPOSX1_21/CLK to DFFPOSX1_24/D	413.983	469.93
3	Maximum Delay	DFFPOSX1_23/CLK to DFFPOSX1_4/D	989.000	1011.12
3	Minimum Delay	DFFPOSX1_21/CLK to DFFPOSX1_24/D	339.647	10.11.12

From the STA analysis carried out it is observed that scheme 3 has minimum propagation delay and is faster in encoding and decoding process as compared with other two schemes. The number of gate count is increased by factor of 10.97% compared with two other schemes. Table 6 presents the power dissipation reports compared for all three different schemes. The static power dissipation in scheme 3 is the lowest and with regard to dynamic power dissipation scheme 2 is the best. Considering trade-off between power-area-delay scheme 3 is recommended for data encoding and decoding.

Table 6. Power dissipation report

Method	Static power (nW)	Dynamic power (nW)
Scheme 1	12.0562943	1.160
Scheme 2	11.8750746	833.37
Scheme 3	9.831569	1048.48

Table 7 presents the area metrics of all three schemes considering standard cells and total core area. It is observed that after placement and routing the number of standard cells has been increased by 4% in all three schemes. The total core area of scheme 3 is higher by 12% as compared with all other schemes.

The cell height and cell width for scheme is 1.39x6.78 units for scheme 3 encoder-decoder logic and the total number of cells of 210 are placed within the area of 3350 units. The propagation delay, area and power dissipation can be further optimized by considering low power strategies. Table 8 compares the performances of the proposed design with the design reported in literature.

Table 7. Area metrics of encode-decoder

Parameter	Scheme 1	Scheme 2	Scheme 3
Total std. cells	290	291	281
Total cell width	6.89e ⁺⁰⁵	6.90e ⁺⁰⁵	6.78e ⁺⁰⁵
Total cell height	1.43e ⁺⁰⁶	1.44e ⁺⁰⁶	1.39e ⁺⁰⁶
Total cell area	3.40e ⁺⁰⁹	3.41e ⁺⁰⁹	3.35e ⁺⁰⁹
Total core area	3.40e ⁺⁰⁹	3.41e ⁺⁰⁹	3.35e ⁺⁰⁹
Avg. height	4.94e ⁺⁰³	4.94e ⁺⁰³	4.94e ⁺⁰³

Table 8. Performance comparison of proposed design

Parameter	Area (μm^2)	Latency(pS)	Power	Frequency
Reference paper [20]	2144	1085	-	68.1 MHz
IMP-FIBO [27]	973	-	178 μW	-
ACME [25]	235	-	88 mW	100 kHz
BCH [28]	-	258	0.787 W	440 MHz
Proposed work	3490	989	1.160 nW	1011.12 MHz

The operating frequency of the proposed design is 15 times faster than the reference design with overhead increase in area by 38%. The overall power reduction in the proposed design is 98% demonstrating the advantages of the encoding scheme for reducing power dissipation over the interconnects. The logic design presented in this work can be further optimized considering pipelined and parallel processing schemes.

5. CONCLUSION

Interconnects play an important role in VLSI circuits by connecting two logic circuits for data transfer. Power dissipation in interconnects have increased due to the requirements of high speed and high data rate requirements. Power dissipation reduction methods are focused on replacing interconnects with low resistive materials. Another approach for interconnect power reduction is the use of encoding schemes for data transmission. In this paper, three different schemes are presented along with design details, synthesis report and physical design implementation. The power dissipation in the proposed encoding schemes is reduced by limiting self-transition and cross coupling transitions. The area, power and delay metrics of encoder-decoder logic is also optimized considering ASIC implementation. The proposed schemes are recommended for low power applications.





REFERENCES

- [1] A. Carpenter, J. Hu, J. Xu, M. Huang, H. Wu, and P. Liu, "Using transmission lines for global on-chip communication," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 183–193, Jun. 2012, doi: 10.1109/JETCAS.2012.2193519.
- [2] P. Dong, Y. K. Chen, T. Gu, L. L. Buhl, D. T. Neilson, and J. H. Sinsky, "Reconfigurable 100 Gb/s silicon photonic network-on-chip [invited]," *Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A37–A43, Jan. 2015, doi: 10.1364/JOCN.7.000A37.
- [3] A. J. Karkar *et al.*, "Hybrid wire-surface wave interconnects for next-generation networks-on-chip," *IET Computers and Digital Techniques*, vol. 7, no. 6, pp. 294–303, Nov. 2013, doi: 10.1049/iet-cdt.2013.0030.
- [4] F. Al Faisal, M. M. H. Rahman, and Y. Inoguchi, "Power analysis with variable traffic loads for next generation interconnection networks," in *Proceedings - 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, Dec. 2017, pp. 422–429, doi: 10.1109/HPCC-SmartCity-DSS.2016.0067.
- [5] M. N. Bojnordi and E. Ipek, "DESC: Energy-efficient data exchange using synchronized counters," in *MICRO 2013 - Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, 2013, pp. 234–246, doi: 10.1145/2540708.2540729.
- [6] G. Pekhimenko *et al.*, "Linearly compressed pages: a low-complexity, low-latency main memory compression framework," in *MICRO 2013 - Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, 2013, pp. 172–184, doi: 10.1145/2540708.2540724.
- [7] A. Shafiee, M. Taassori, R. Balasubramanian, and A. Davis, "MemZip: Exploring unconventional benefits from memory compression," in *Proceedings - International Symposium on High-Performance Computer Architecture*, Feb. 2014, pp. 638–649, doi: 10.1109/HPCA.2014.6835972.
- [8] G. Pekhimenko, E. Bolotin, M. O'Connor, O. Mutlu, T. C. Mowry, and S. W. Keckler, "Toggle-aware compression for GPUs," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 164–168, Jul. 2015, doi: 10.1109/LCA.2015.2430853.
- [9] A. Garcia-Ortiz, D. Gregorek, and C. Osewold, "Optimization of interconnect architectures through coding: A review," in *2011 Saudi International Electronics, Communications and Photonics Conference (SIEPCPC)*, Apr. 2011, pp. 1–6, doi: 10.1109/SIEPCPC.2011.5876688.
- [10] V. Adhinarayanan, I. Paul, J. L. Greathouse, W. Huang, A. Pattnaik, and W. Feng, "Measuring and modeling on-chip interconnect power on real hardware," in *2016 IEEE International Symposium on Workload Characterization (IISWC)*, Sep. 2016, pp. 1–11, doi: 10.1109/IISWC.2016.7581263.
- [11] Y. Wang and H. Wu, "Design high bandwidth-density, low latency and energy efficient on-chip interconnect," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Jul. 2017, pp. 1–6, doi: 10.1109/ISLPED.2017.8009171.
- [12] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 264–272, Jun. 2000, doi: 10.1109/92.845893.
- [13] M. W. Chaudhary, A. Heinig, and B. Choubey, "Interconnect aware power optimization of low swing driver for multi-chip interfaces," in *ICECS 2020 - 27th IEEE International Conference on Electronics, Circuits and Systems, Proceedings*, Nov. 2020, pp. 1–4, doi: 10.1109/ICECS49266.2020.9294796.
- [14] J. C. Garcia, J. A. Montiel-Nelson, and S. Nooshabadi, "Adaptive low/high voltage swing CMOS driver for on-chip interconnects," in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 881–884, doi: 10.1109/ISCAS.2007.378047.
- [15] J. Postman and P. Chiang, "A survey addressing on-chip interconnect: energy and reliability considerations," *International Scholarly Research Notices*, vol. 2012, pp. 1–9, Mar. 2012, doi: 10.5402/2012/916259.
- [16] M. Chennakesavulu, T. Jayachandra Prasad, and V. Sumalatha, "Data encoding techniques to improve the performance of system on chip," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 492–503, Feb. 2022, doi: 10.1016/j.jksuci.2018.12.003.
- [17] H. Y. To, "An analysis of data bus inversion: examining its impact on supply voltage and single-ended signals," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 2, pp. 31–41, 2019, doi: 10.1109/mssc.2019.2910626.
- [18] E. Maragkoudaki, P. Mrosczyk, and V. F. Pavlidis, "Adaptive word reordering for low-power inter-chip communication," in *Proceedings of the 2019 Design, Automation and Test in Europe Conference and Exhibition, DATE 2019*, Mar. 2019, pp. 980–983, doi: 10.23919/DATE.2019.8714820.
- [19] E. Maragkoudaki and V. F. Pavlidis, "Energy-efficient time-based adaptive encoding for off-chip communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 12, pp. 2551–2562, Dec. 2020, doi: 10.1109/TVLSI.2020.3018062.
- [20] K. Kim, S. Kang, and B. Kim, "A code inversion encoding technique to improve read margin of a cross-point phase change memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1811–1818, Aug. 2019, doi: 10.1109/TVLSI.2019.2909535.
- [21] M. Bhushan and M. B. Ketchen, *CMOS test and evaluation: A physical perspective*. New York, NY: Springer New York, 2015.





- [22] S. K. Vendra and M. Chrzanowska-Jeske, "Buffered-interconnect performance and power dissipation in 3D ICs with temperature profile," in *Proceedings - IEEE International Symposium on Circuits and Systems*, May 2018, vol. 2018-May, pp. 1–5, doi: 10.1109/ISCAS.2018.8351416.
- [23] M. R. Stan and W. P. Burlison, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 1, pp. 49–58, Mar. 1995, doi: 10.1109/92.365453.
- [24] M. Yoon, "A two-bit bus-invert coding scheme with a mid-level state bus-line for low power VLSI design," *Journal of Semiconductor Technology and Science*, vol. 14, no. 4, pp. 436–442, Aug. 2014, doi: 10.5573/JSTS.2014.14.4.436.
- [25] C. Xie, D. J. Pagliari, A. Calimera, E. Macii, and M. Poncino, "ACME: An energy-efficient approximate busencoding for I2C," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Jul. 2021, pp. 1–6, doi: 10.1109/ISLPED52811.2021.9502495.
- [26] S. Sangewar, "Encoding scheme for power reduction in network on chip links," in *7th International Conference on Computing in Engineering & Technology (ICET 2022)*, 2022, pp. 347–351, doi: 10.1049/icp.2022.0645.
- [27] B. Subramaniam, S. Muthusamy, and G. Gengavel, "Crosstalk minimization in network on chip (NoC) links with dual binary weighted code CODEC," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 4603–4608, May 2021, doi: 10.1007/s12652-020-01842-1.
- [28] W. U. Tao and Y. A. N. G. Ailin, "A simple BCH decoder for NoC interconnects and SoC buses," *Chinese Journal of Electronics*, vol. 30, no. 3, pp. 444–450, May 2021, doi: 10.1049/cje.2021.03.007.

BIOGRAPHIES OF AUTHORS







Vennapusapalli Shavali     is a Research scholar, pursuing Ph.D in VLSI from Jawaharlal Nehru Technological University Anantapur (JNTUA)Anantapur, India. Completed M.Tech in Digital electronics and communication from JNTUA. His field of interest in electronics circuits, analog circuits, digital circuits and all integrated circuits. He can be contacted at email: v.shavali@gmail.com.



Prof. Dr. Sreeramareddy Gorlagummanahally Maripareddy     completed Ph.D from Jawaharlal Nehru Technological University Anantapur (JNTUA), Anantaour, India. He is interested in research activities in VLSI, image processing, optical networks. He motivated many students and scholars to do research. He is guiding ph.D scholars. He worked in many administrative positions as HOD, Dean of Academics, Principal, BOS member, Academic senate member in VTU, Karnataka, India. He published many papers in conferences and Journals. He can be contacted at email: sreeramareddy90@gmail.com.



Prof. Dr. Patil Ramana Reddy     is a working as professor in Dept. of ECE, Jawaharlal Nehru Technological University Anantapur (JNTUA), Anantapur, A. P, India. He Completed his Ph.D. from JNTUA, Anantapur. He worked in many Administrative positions like HOD, ECE, JNTUA, CEA, Head of ECE & vice principal JNTUA CEK, Faculty advisors, BOS chair persons. He motivated many students to perusing Ph.D. He guided many scholars and published many papers in national, international conference and publications. He can be contacted at email: prjntu@gmail.com.